

A Creativity Survey of Fully Dynamic Maximal Independent Set in Expected Poly-log Update Time

TIANXIN LIANG

Department of Computer Science, University of Chinese Academy of Sciences;
liangtianxin21@mails.ucas.ac.cn

WEI LI

Institute of Computing Technology, Chinese Academy of Sciences;
liwei@ict.ac.cn

Abstract: This paper focus on the researches of **Maximal Independent Set (MIS)**. Based on reading and analysis of several recent papers, we divide the MIS problems into several classifications. The first is the classification based on the **research objects**, including the solution and maintenance of MIS; the second is the classification based on **research methods**, including serial, parallel, deterministic and randomized algorithms; the third is **experimental analysis**, including worst time complexity and expected time complexity.

Keywords: maximal independent set, classification, algorithm, theoretical computer science, time complexity

1 INTRODUCTION

A maximal independent set (MIS) of a given graph $G = (V, E)$ is a subset M of vertices such that M does not contain two neighbor vertices and every vertex in $\frac{V}{M}$ has a neighbor vertex in M .

MIS is a fundamental problem in the field of computer science with both theoretical and practical significance. In a multi-core environment, parallel threads are widely used in resource scheduling, election, feature tracking and so on. It has been studied for decades how to find MIS efficiently. Even in early 1980s, the efficient parallel algorithm is presented. With the ever-increasing amount of information and user demands, the dynamic MIS problem has been widely studied in recent years since 2010.

We select a recent paper about research of dynamic MIS algorithm (reference [1]). Based on reading and analysis of several relative references about the MIS problem, we divide the MIS problem into several classifications. Our result shows the research focus and innovation of each reference, and divides them into discrete types by research objects, research methods and experimental analysis.

In section 2, we divide the references based on the **research objects**, including the solution and maintenance of MIS; in section 3, we divide them based on **research methods**, including serial, parallel, deterministic and randomized algorithms; and in section 4, we divide them based on **experimental analysis**, including worst time complexity and expected time complexity.

The rest of the paper is organized as follows. Section II gives the classification of research objects of MIS problems. Section III introduces the classification of research methods. Section IV introduces the comparison

of experimental analysis in related literature. Section V discusses the research opportunities in future work and Section VI concludes the paper.

2 CLASSIFICATION OF RESEARCH OBJECTS

Table1: Different Research Objects

Parallel Sorting Algorithm	Problem Type	
	MIS finding	MIS maintaining
Theory Research	I. [2][12]	II. [1][3][4][5][7][9][10][13]
Algorithm Application	III. [8][11][14]	IV. [8]

2.1 Criteria

As a basic model in computer theory, MIS problem is not only of theoretical research significance, but also has important application value in practical problems. For the MIS that has been obtained, if the local environment changes, we also hope to construct a new MIS in the shortest possible time. In this section, two independent and different criteria would be used to divide research objects into different types:

1) **Aspect.** There are two types here: **Theory Research** or **Algorithm Application**. Theory Research aims to study the general algorithms used to solve abstract mathematical models, focusing on the correctness and complexity of the algorithms themselves. Algorithm Application is to establish a suitable mathematical model for practical problems and transform them into MIS problems, focusing on the application of algorithms in practical problems.

2) **Problem Type.** There are two kinds of problems here: **MIS finding (Static MIS)** or **MIS maintaining (Dynamic MIS)**. MIS finding Aims to study how to efficiently find an MIS in an undirected graph. Since the serial algorithm has the general $O(n)$ time complexity, the parallel algorithm is usually studied. MIS maintaining aims to study how to find a new MIS when the relationships between nodes change after the MIS is obtained.

2.2 The Classification

Based on the appeal classification standard, we give the classification in Table 1. The meaning of each class is as follows:

Type I: This type is to study that, given an undirected graph, with almost no additional information, how to efficiently find a maximal independent set in a short time, called "Static MIS".

Type II: This type is to study that, on the premise that a maximal independent set has been found, when the local environment is disturbed (such as the addition/deletion of a node or edge), how to use the current information to efficiently find a new maximal independent set, called " Dynamic MIS".

Type III: This type is to study the application of Static MIS algorithm in practical problems.

Type IV: This type is to study the application of Dynamic MIS algorithm in practical problems.

2.3 Explanation of Different Types

References ([2][12]) belong to Type I. Reference [2] finds MIS in parallel environments using $O(|E|d_{max})$ processors. Reference [12] uses two strategies to solve MIS problem, by assigning identical copies of a

simple algorithm to small local portions or converting the Monte Carlo algorithm into a simple deterministic algorithm.

References ([1][3][4][5][7][9][10][13]) belong to Type II. Reference [1] is the target paper. Reference [7] studies the MIS problem in distributed and proposes the concept of dynamic MIS. In reference [3], the first algorithm for dynamically maintaining MIS in sublinear time is given. For the cases of adding edges, deleting edges, and modifying nodes, a MIS can be found again in sublinear time. Reference [9][10][4][13][5] improve the dynamic MIS algorithm respectively, and gradually reduce the time complexity of the algorithm.

References ([8][11][14]) belong to Type III. Reference [8] introduces the application of MIS in leader election. Reference [14] introduces the application of MIS in resource allocation. Reference [11] introduces the application of MIS in distributed graph algorithms.

References ([8]) belong to Type IV. Reference [8] involves the impact of dynamically modifying MIS in leader election.

3 CLASSIFICATION OF RESEARCH METHODS

Table 2: Different Research Methods

Implement	Randomization	
	Deterministic	Randomized
Serial	I.[4][9][10][13]	II. [1][4][5][6][9]
Parallel	III. [12]	IV. [2]

3.1 Criteria

In order to overcome the bottleneck of the serial running time of the algorithm, it is often necessary to use a parallel algorithm to run in multiple threads at the same time to achieve the effect of acceleration, while the parallel algorithm and the serial algorithm are quite different in design ideas, except for the algorithm itself. In addition to the complexity of each thread, the execution order and synchronization between threads also need to be considered. In addition, some algorithms are random and have superior robustness and generalization ability, but the analysis of performance is also more complicated than that of deterministic algorithms. Therefore, in this section, two independent and different criteria would be used to divide research objects into different types:

1) **Implement.** There are two types here: **Serial** or **Parallel**. A serial algorithm is an algorithm in which instructions are executed sequentially on a single processor, while a parallel algorithm is an algorithm in which threads are executed synchronously on multiple processors.

2) **Randomization.** There are two kinds of attribute here: **Deterministic** or **Randomized**. Deterministic means the algorithm will run in the same way if the inputs are identical. Randomized means for identical inputs, the algorithm may run in different ways.

3.2 The Classification

Based on the appeal classification standard, we give the classification in Table 2. The meaning of each class is as follows:

Type I: This type is serial deterministic algorithm, means the deterministic algorithm that runs on a single-core processor.

Type II: This type is serial randomized algorithm, means the randomized algorithm that runs on a single-core processor.

Type III: This type is parallel deterministic algorithm, means the deterministic algorithm that runs on a multi-core processor.

Type IV: This type is parallel randomized algorithm, means the randomized algorithm that runs on a multi-core processor.

3.3 Explanation of Different Types

References ([3][9][10][13]) belong to Type I. They each present at least one deterministic algorithm that can be run in serial environments.

References ([1][4][5][6][9]) belong to Type II. They each present at least one randomized algorithm that can be run in serial environments.

References ([12]) belong to Type III. It presents a deterministic algorithm that can be run in parallel environments.

References ([2]) belong to Type IV. It presents a randomized algorithm that can be run in parallel environments.

4 REVIEW OF EXPERIMENTAL ANALYSIS

In this section, we will classify the metric of evaluation and system parameters, as shown in Table 3. In Table 3, all experimental analysis is also classified according to the metric and parameters. It can be seen from Table 3 that most of the references compare **Worst-case time complexity** and **Expected time complexity**.

Table 3: Experiments with Different Metric and Parameters

Metric	Parameters	
	Static MIS	Dynamic MIS
Worst-case time complexity	I.[12]	II.[1][5][7][9]
Expected time complexity	III.[2]	IV.[1][3][4][5][6][9][10][13]

4.1 Metric of Evaluation

Worst-case time complexity means the worst time complexity among all cases. The formula is as follows:

$$\text{Worst - case time complexity} = \text{Max}\{\text{running time complexity in a certain case}\}$$

Expected time complexity means the average time complexity of all cases. The formula is as follows:

$$\text{Expected time complexity} = \frac{\sum \text{running time complexity in a certain case}}{\text{Number of total cases}}$$

4.2 System Parameters

Static MIS represents the evaluation of **MIS finding**. MIS finding Aims to study how to efficiently find an MIS in an undirected graph.

Dynamic MIS represents the evaluation of **MIS maintaining**. MIS maintaining aims to study how to find a new MIS when the relationships between nodes change after the MIS is obtained.

4.3 Experimental Comparison

In reference [7], the author studies the problem of maintaining an MIS in the distributed dynamic setting where the graph changes over time, and asks if there is an algorithm running faster than recomputing.

In reference [3], the author presents a $O^{\sim}(\min\{\Delta, m^{\frac{3}{4}}\})$ amortized expected time deterministic algorithm to maintain MIS, which is the first algorithm maintaining MIS in sublinear time complexity.

In reference [9], the author improves the worst-cast time complexity to $O(\min\{\Delta, m^{\frac{2}{3}}\sqrt{\log m}\})$, and presents a randomized algorithm in $O(\min\{\Delta, \sqrt{m} \log^{1.5} m\})$ expected time complexity.

In reference [10], the author presents an algorithm in $O(\min\{\Delta, m^{\frac{2}{3}}\sqrt{\log m}\})$ independently with [9].

In reference [4], the expected time complexity is further improved to $O^{\sim}(\sqrt{n})$ and $O^{\sim}(m^{\frac{1}{3}})$.

In reference [13], the author improves the expected time complexity to $O(\alpha^2 \cdot \log^2 n)$ on uniformly sparse graphs.

In reference [5], the author presents a polynomial-time algorithm in $O(\log^2 \Delta \cdot \log^2 n)$ expected time complexity and $O(\log^2 \Delta \cdot \log^4 n)$ worst-cast time complexity.

In reference [2], the author presents a parallel algorithm for solving a maximal independent set on $O(|E|d_{\max})$ processors with $O(\log n)$ expected time complexity.

In reference [12], a parallel algorithm for solving a maximal independent set on $O(n^2 m)$ processors with $O(\log^2 n)$ worst-time complexity is proposed.

5 DISCUSSION AND SUGGESTIONS

This paper discusses the research methods and research objects of various references and finds that most of the papers study the algorithm theory of dynamic MIS problem, and they are most about the gradual optimization of serial complexity, while the research on optimization of other indicators and the application of dynamic MIS are few. Therefore, this paper puts forward the following directions, which can provide directions for future research of MIS problem:

- 1) Optimize the constant and space complexity of dynamic MIS algorithm. Most of the above papers optimize the time complexity of the algorithm, without focusing on the actual running time and space complexity.
- 2) Integrate the existing dynamic MIS algorithm into specific application environments. With the theoretical basis of the algorithm, we can consider combining it with practical applications and examine its performance in practical applications.
- 3) Research the parallel algorithm of dynamic MIS. Nowadays, the use of multi-core environments is much extensive. With efficient parallel algorithm, the time complexity bottleneck of serial algorithm may be broken.

6 CONCLUSIONS

Through the previous analysis, it is found that the time complexity of dynamic MIS algorithm is constantly improving in recent years; however, there are few studies on the application of the algorithm in specific environments and other aspect. Therefore, this paper puts forward some suggestions for possible future innovative work.

The limitation of our work is that we have not yet delved into the algorithmic principles of these papers. Through further fine-grained analysis and division of the theorems and data structures used by each algorithm, it may be possible to find and combine the advantages of each algorithm, thereby discovering new innovations.

References

- [1] Chechik S, Zhang T. Fully dynamic maximal independent set in expected poly-log update time[C]//2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS). IEEE, 2019: 370-381.
- [2] Itai A, Babai L, Alon N, A fast and simple randomized parallel algorithm for the maximal independent set problem[J]. Journal of algorithms, 1986, 7(4): 567-583.
- [3] Sepehr Assadi, Krzysztof Onak, Baruch Schieber, and Shay Solomon. Fully dynamic maximal independent set with sublinear update time. In Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018, pages 815–826, 2018.
- [4] Schieber B, Onak K, Assadi S, et al. Fully dynamic maximal independent set with sublinear in n update time[C]//Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 2019: 1919-1936.
- [5] Derakhshan M, Behnezhad S, Hajiaghayi M T, et al. Fully dynamic maximal independent set with polylogarithmic update time[C]//2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS). IEEE, 2019: 382-405.
- [6] Bernstein A, Henzinger M, Forster S. A deamortization approach for dynamic spanner and dynamic maximal matching[J]. ACM Transactions on Algorithms (TALG), 2021, 17(4): 1-51.
- [7] Haramaty E, Censor-Hillel K, Karnin Z. Optimal dynamic distributed MIS[C]//Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing. 2016: 217-226.
- [8] Kuhn F, Daum S, Gilbert S, et al. Leader election in shared spectrum radio networks[C]//Proceedings of the 2012 ACM symposium on Principles of distributed computing. 2012: 215-224.
- [9] Zhang H , Du Y. Improved algorithms for fully dynamic maximal independent set[J]. arXiv preprint arXiv:1804.08908, 2018.
- [10] Gupta M, Khan S. Simple dynamic algorithms for maximal independent set and other problems[J]. arXiv preprint arXiv:1804.01823, 2018.
- [11] Linial N. Distributive graph algorithms global solutions from local data[C]//28th Annual Symposium on Foundations of Computer Science (sfcs 1987). IEEE, 1987: 331-335.
- [12] Luby M. A simple parallel algorithm for the maximal independent set problem[J]. SIAM journal on computing, 1986, 15(4): 1036-1053.
- [13] Schieber B, Onak K, Solomon S, et al. Fully dynamic MIS in uniformly sparse graphs[J]. arXiv preprint arXiv:1808.10316, 2018.
- [14] Yu D, Hua Q S, Wang Y, et al. Distributed $(\Delta+1)$ -coloring in the physical model[J]. Theoretical Computer Science, 2014, 553: 37-56.